

FSA Integration Partner

United States Department of Education

Federal Student Aid



FAFSA 7.0 Detailed Performance Environment Configuration

Task Order #142

Deliverable 142.1.3

Version 1.0

September 5, 2003



Executive Summary

Introduction

The U.S. Department of Education's Office of Federal Student Aid (FSA) administers and operates the "Free Application for Federal Student Aid" (FAFSA). While available in paper form, FSA also provides this service through a web site. U.S. college students seeking student financial aid use the FAFSA program. Starting from May, 2003, FSA had been under taking the task of upgrading its application server software from IBM WebSphere Application Server (WAS) v.3.5.6 to v.5.0. The upgrade is part of FSA ever greening work and it will allow FSA to take advantage of the new features and performance enhancements offered by WAS 5.0. As part of the process, applications that are running on WAS 3.5.6 will need to be validated against J2EE 1.3 specifications and packaged for deployment. For FAFSA 7.0, this re-platforming process will need to be verified in order to ensure it is working properly.

Background

FAFSA 7.0 went live on January 1, 2003. FAFSA 7.0 is unique among FSA applications with its support of a 6 million user base. On the peak day, one hundred and one thousand application submissions were recorded. Architecturally, FAFSA 7.0 is hosted on HP servers utilizing WebSphere and IBM HTTP Server (IHS). The application connects to backend Oracle and DB2 databases. IBM WebSphere MQ and CICS are also used for enterprise messaging and transaction management.

Objective

The purpose of the FAFSA 7.0 on WAS 5.0 performance test was to validate that the FAFSA WebSphere 5.0 configuration could withstand a load of 1000 users submitting and correcting FAFSA applications.



Amendment History

DATE	SECTION / PAGE	DESCRIPTION	Submitted BY
09/03/03	All	Performance test configuration document	Roshani Bhatt



Table of Contents

1	INTRODUCTION	5
1.1	OVERVIEW	5
1.2	FAFSA SOFTWARE MATRIX	5
1.3	COMPONENT ARCHITECTURE.....	5
1.3.1	<i>Load Balancer</i>	<i>5</i>
1.3.2	<i>Web Server.....</i>	<i>5</i>
1.3.3	<i>Application Server</i>	<i>6</i>
1.3.4	<i>Messaging.....</i>	<i>6</i>
1.3.5	<i>Operating System.....</i>	<i>6</i>
1.3.6	<i>Security.....</i>	<i>6</i>
1.4	FAFSA ON THE WEB CURRENT CONFIGURATION.....	7
2	IBM BEST PRACTICES	8
2.1	WEB APPLICATION - RELOAD INTERVAL AND ENABLE	8
2.2	WEB CONTAINER MAXKEEPALIVECONNECTIONS AND MAXKEEPALIVEREQUESTS	8
2.3	SESSION SIZE	8
2.4	SESSION AFFINITY	9
2.5	MAX CLIENTS	9
2.6	REFRESHINTERVAL	9
3	PERFORMANCE ENVIRONMENT CONFIGURATION.....	10
3.1	WEB SERVERS.....	10
3.2	APPLICATION SERVER.....	10
3.2.1	<i>Template Application Server parameters</i>	<i>11</i>
3.3	OTHER SCRIPTS.....	12
3.4	OTHER SOFTWARE (RCS, WILY, WSADMIN)	12
3.5	SHARED LIBRARIES.....	13
3.5.1	<i>JDBC Providers.....</i>	<i>13</i>
3.5.2	<i>Oracle JDBC</i>	<i>14</i>
3.5.3	<i>Shadow Direct JDBC.....</i>	<i>14</i>
4	APPENDIX – FAFSA PRODUCTION ARCHITECTURE	16
4.1	LOAD BALANCING	16
4.2	WEB SERVERS.....	16
4.3	APPLICATION SERVERS	16
4.4	MAINFRAME	17
4.5	DATABASE.....	17



1 Introduction

1.1 Overview

This document outlines the configuration of the performance environment **for FAFSA on the Web 7.0** application. This environment will be used to stress test the performance environment Infrastructure as well as the FAFSA application code. The environment is composed of one Sun 220 Solaris 2.8 server that will be used as a load balancer and Network deployment manager. Two HP L-class HP-UX 11.0i servers will serve as web servers and 2 HP-UX 11i N-class servers will server as WebSphere application servers. The following table specifies the ITA software versions.

1.2 FAFSA software matrix

The following matrix captures the components of FAFSA that have been updated from the application's January 2003 release to August.

No.	Component	Version	System
1	Operating System	HP-UX 11i	HPL14, HPL17, HPN3, HPN8
2	Edge Server 2.0 (Network Dispatcher) WebSphere Deployment Manager 5.01	4.0.25 5.01	SU22e20 SU22e24
3	IBM HTTP Server	1.3.26.1	HPL14, HPL17
4	WebSphere 5.01 Network Deployment +efix PQ77056	5.0.1	HPN3, HPN8
5	WebSphere MQ (on application server)	5.3.1	HPN3, HPN8

1.3 Component Architecture

1.3.1 Load Balancer

Edge Server is used to load balance HTTP requests to the web servers. Network Dispatcher includes HTTP advisors that alert the load-balancing component of Edge Server if a Web Server is down. Edge server is configured with its high availability component which means that if the primary Edge Server goes down or is taken down, the backup Edge Server immediately comes online.

1.3.2 Web Server

IBM HTTP Server (IHS) is an Apache based web server that provides HTTP server functionality. Any static HTTP documents can be served via the Web Server. The IHS Web Server plugin provides connectivity from the web server to the application server. The plugin



provides load balancing and failover for the application server. Each web server can connect to both Application Servers.

1.3.3 Application Server

WebSphere is a Java Based Servlet and JSP Engine that conforms to Sun Microsystems's J2EE specification. WebSphere is built upon the Java Development Kit 1.3.1.08, which includes servlet specification 2.3 and JSP specification 1.2.

1.3.4 Messaging

An MQSeries cluster is configured on the application servers so that the FAFSA application can drop messages to the EAI cluster, which immediately transports it to the destination.

The following diagram shows a pictorial representation of what the performance environment looks like.

1.3.5 Operating System

HP-UX Version 11.0i on Web and App Servers
December 2002 Quality Patch

Ndd parameters – /etc/rc.config.d/nddconf

tcp_conn_request_max 1024

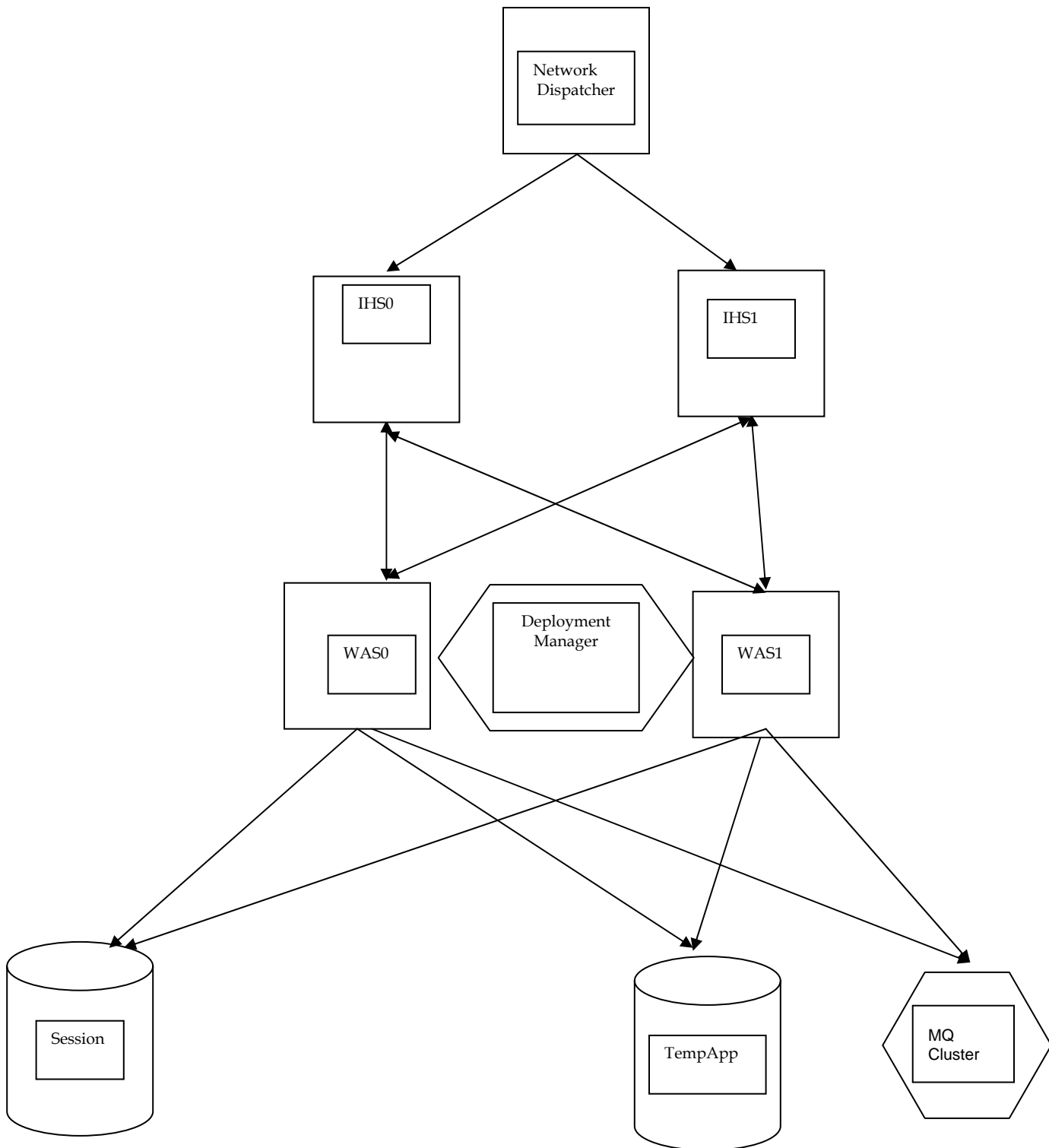
1.3.6 Security

Admin Console Security was enabled for the performance test. The use of Computing Base Security by CSC on the HP-UX systems made the local OS registry incompatible with WebSphere security. Also since no LDAP registry is available, the only other option is to use a custom registry that comes with WebSphere. Basically User IDs and passwords are stored in root owned files and WebSphere verifies User ID and Passwords using built in custom registry support. The password and User ID files are located at:

/opt/WebSphere/security/users.props
/opt/WebSphere/security/groups.props



1.4 FAFSA on the Web Current Configuration





2 IBM Best Practices

2.1 Web application - Reload interval and enable

Reload enable specifies whether the Web application's file system is scanned for updated files periodically on the Application Server. If files such as servlet class files or JSPs are found to be updated since the last scan then they can be reloaded. Reload enable should be turned off for production.

[Applications](#)-> [Enterprise Applications](#)-> [FOTW App](#) -> Reload Enabled

2.2 Web Container MaxKeepAliveConnections and MaxKeepAliveRequests

Since WebSphere 5.0 has introduced HTTP 1.1 protocol between the Web Server and the Application Server, the Web Server plugin now can keep persistent HTTP connections open to the application server for a defined number of requests. This enhances performance due the fact that the pre 5.0 plugin had to open a socket port for every Application Server request. To enable this performance enhancement, the following custom properties must be added to the HTTP Transport configuration. This should be added for the SSL and non SSL port. The MaxKeepAliveConnections for SSL and nonSSL should add to 80% of the maximum thread allowed for the Web Server.

[Application Server](#)-><[Server name](#)>->[Web Container](#)->[HTTP Transport](#)-><[Port number](#)>

MaxKeepAliveConnections	40
MaxKeepAliveRequests	1000

This should be added for the SSL and non SSL port. The MaxKeepAliveConnections for SSL and non-SSL should add to 80% of the maximum thread allowed for the Web Server. So the above configuration assumes a Servlet thread maximum of 100.

[Application Server](#)-><[Server name](#)>->[Web Container](#)->Thread Pool

Minumum Size	0
Maximum Size	100
Thread inactivity timeout	3500

2.3 Session Size

The two business processes that were used during the performance test have been checked to ensure that persisted HTTP servlet session size is less then 4K. The FAFSA business processes and their respective session size appear in the table below.

Fill Out a FAFSA	2.9 Kbytes
Corrections	4.1 Kbytes

This is checked using a JSP file called SessionObjects.jsp.



2.4 *Session Affinity*

Due to the fact that Session Affinity is guaranteed by the Web Server plugin on the client's first request, a Session database is not considered critical to ensuring the client's request goes to the same application server every time. This simple enhancement turns the session database into a failover device and allows WebSphere to reduce the number of session database writes and reads. In fact, with WebSphere 5.0 the session management configuration has reduced session database writes from every servlet call (3.5.6) to a time based system. The Application Server session management needs to be configured for the custom setting (default). This allows only updated attributes to be saved and the session is saved every 120 seconds.

[Application Server](#)-><[Server name](#)>-><[Web Container](#)>-><[Session Management](#)>->Distributed Environments Settings

Due to the fact that the Session information is written to the database so much less, this requires less JDBC connections to be used for session writes. Since the footprint of each JDBC connection is 1 to 2MB, this translates to a smaller heap (512MB) and a much faster garbage collection.

2.5 *Max Clients*

The value of the IHS max client parameter can significantly affect the performance of an application. If this value is too high, it is possible to overwhelm the application server and cause it to stall. If this value is too low then it is possible to induce a bottleneck with client requests before they reach the application server. Implementing the performance test using 1000 users that submit approximately 50-70 FAFSA applications a minute, the performance test found that the number of HTTP daemons never rose above 400 with one web server. Thus the recommended max client setting for two web servers would be 300.

2.6 *RefreshInterval*

This parameter sits in the IHS plugin file and determines the amount of time elapsed before IHS will reread the plugin data. This parameter is very useful to enable administrators to quiesce (temporarily deactivate) Application Servers. By editing and changing the plugin file, IHS will reread the file in <RefreshInterval> seconds. Thus if a server is weighted=0 in plugin, IHS will stop sending traffic as soon as it rereads the plugin. The default with no parameter specified is 60 seconds.



3 Performance Environment Configuration

3.1 Web Servers

IBM HTTP Server 1.3.26.1 + efix pq75807

Pq75807 is a change to the HTTP plugin that is HP-UX specific. The original plugin did not pass SSL attribute data correctly to the Application Server. When adding additional plugin efix's or fix pacs, special care needs to be taken to ensure that this efix is included.

The following Web Server Configuration is maintained in the
/opt/HTTPServer/conf/HTTPd.conf file on both web servers.

MaxClients	300
MinSpareServers	10
MaxSpareServers	100
StartServers	10
MaxRequestPerChild	10000
MaxKeepAliveRequests	100
KeepAliveTimeout	15

Non SSL Name Based Virtual Hosts

Url	HTTP://perf.fotw.ed.gov
Doc root	/www/fotw/htdocs
IP address	198.77.203.79
Port	80

SSL Name Based Virtual Host

Url	HTTPS://perf.fotw.ed.gov
Doc root	/www/fotw/htdocs
IP address	198.77.203.79
Port	443
Certificate	Fotwperf

3.2 Application Server

Version 5.01 +PQ77056

PQ77056 is a efix added to fix pac 5.01 to allow storage of session data greater then 4k to a oracle database.

The performance test used a cluster of four Application Servers. Two Application servers resided on HPN8 following the naming convention of FOTWn8c1 and FOTWn8c2. Two



Application Servers resided on HPN3 following the naming convention FOTWn3c1 and FOTWn3c2. The cluster is called FOTWCluster. It is important to keep in mind when building out a Cluster that building an initial non-clustered Application Server template and then building out clustered AppServers using the template. To ensure completeness, the above performance parameters are repeated in this section.

3.2.1 Template Application Server parameters

The current version of WAS employs a web-based administrative client that may viewed in a browser, but the previous WAS version (3.5.6) utilized a Java client. The path to the “Java Virtual Machine” is included here as it appears in the WAS 5.0 administrative client to illustrate this new capability.

[Application Servers](#) > [FOTWServer](#) > [Process Definition](#) > Java Virtual machine

```
Classpath
BootClasspath
Initial Heap Size =512
Maximuml Heap Size =512
Verbose garbage collection=checked (Gc data goes to native_stdout.log)
```

[Application Servers](#) > [FOTWServer](#) > [Process Definition](#) > [Java Virtual Machine](#) >Custom Properties

```
APP_PROPERTY_PATH = ${FOTW_ROOT}/properties
Autonomy.config.xml = ${FOTW_ROOT}/properties/autonomy.properties
EAI_CONNECTION_MODEL = POOL
EAI_MQ_PROP_FILE = /www/dev50/eai/eaicodr1/config/mqipool.properties
LOGGING_CONFIG_FILE = www/dev50/eai/eaicodr1/config/EAICONF.INI
Syslog.config.xml = ${FOTW_ROOT}/properties/rcs.xml
awt.toolkit = com.eteks.awt.PJAToolkit
com.ibm.websphere.sendredirect.compatibility = true
java.awt.fonts = ${JAVA_HOME}/jre/lib/fonts
java.awt.graphicsenv = com.eteks.java2d.PJAGraphicsEnvironment
java2d.font.usePlatformFont = false
log4j.configuration = file:${FOTW_ROOT}/properties/log4j.properties
user.home = ${FOTW_ROOT}/lib/pja
```

[Application Servers](#) > [FOTWServer](#) > [Process Definition](#) > HTTP Transport >Port Number >Custom Properties

MaxKeepAliveConnections	40
MaxKeepAliveRequests	1000

[Application Server](#)-><[Server name](#)>->[Web Container](#)->Thread Pool

Minumum Size	0
Maximum Size	100
Thread inactivity timeout	3500



[Application Server](#)-><[Server name](#)>-><[Web Container](#)>->Session Management

Enable Cookies - Check
Enable URLRewriting - Check
Allow Overflow – Check
Session Timeout = 30 Minutes
Maximum in memory Session Count = 1000

Sessions were set up via the persistent session configuration to store sessions to a database.
Upon startup of the different Application Servers

3.3 Other Scripts

NodeSM – Nanny Startup for the node agent that was written at FSA. This script monitors the Node Agent Java process and restarts it up to 3 times if the node agent dies with a non-zero return code. The Vanialla WebSphere product does not have a monitoring nanny for the node agent. This script calls startNode_mod.sh which must reside in the <WAS_ROOT>/bin directory. NodeSM only needs to reside on nodes that have node agents in the /opt/WebSphere/AppServer/bin directory

An example of starting this would be:

```
cd /opt/WebSphere/AppServer/bin
./NodeSM &
```

ManagerSM - Nanny Startup for the Deployment Manager agent that was written at FSA. This script monitors the Deployment Manager Java process and restarts it up to 3 times if the Deployment Manager dies with a non-zero return code. The Vanialla WebSphere product does not come with a monitoring nanny for the Deployment Manager. This script calls startManager_mod.sh which must reside in the <WAS_DEPLOYMENTMANAGER_ROOT>/bin directory. ManagerSM only needs to reside on the Deployment Manager in the /opt/WebSphere/DeploymentManager/bin directory

For example:

```
Cd /opt/WebSphere/DeploymentManager/bin
./ManagerSM &
```

3.4 Other Software (RCS, Wily, wsadmin)

```
/opt/WebSphere/AppServer/lib/ITA/protomatter-1.1.8.jar
/opt/WebSphere/AppServer/lib/ITA/ rcs.exception.2.0.jar
/opt/WebSphere/AppServer/lib/ITA/ rcs.logging.4.0.jar
/opt/WebSphere/AppServer/lib/ITA/ rcs.search.5.0.jar
/opt/WebSphere/AppServer/lib/ITA/StartupRcs.jar
```



/opt/WebSphere/AppServer/lib/ITA/rcs.persistence.5.0.jar
/opt/WebSphere/AppServer/lib/ext/AutoProbe.jar
/opt/WebSphere/AppServer/lib/ext/Agent.jar
/opt/WebSphere/AppServer/scripts

3.5 Shared Libraries

[Environment](#)->Shared Libraries

Shared libraries are useful when different versions of a common framework are to be associated to different applications. An example of this is when Application A makes use of logging framework 1.2 and Application B that uses logging framework 1.0. The following Jar files are defined as Shared Libraries:

CMNJLOG.jar
EAI.jar
StartupRcs.jar
Com.ibm.mq.jar
Connector.jar
Jakarta-oro-2.0.5.jar
Mqm java lib
Protomatter-1.1.8.jar
Rcs.exception.2.0.jar
Rcs.logging.4.0.jar
Rcs.persistence.5.0.jar
Rcs.search.5.0.jar

Within each Application Server, these jars were added to the Application Classloader

[Application Servers](#)-><[Server Name](#)-><[ClassLoader](#)->ClassLoader_1

3.5.1 JDBC Providers

The Oracle data sources are used in saving Temp Application data, PIN data and Session data. The Shadow JDBC providers just need to be created so that the JNDI can find their names. They are not used in any real database transaction.

During performance testing with 1000 users, One App server never exceeded 6 JDBC connections for sessions or TempApp Storage.

Name	Oracle JDBC Thin Driver	ShadowDirect JDBC Driver
Description	Oracle JDBC Thin Driver	Neon ShadowDirect JDBC driver to connect to DB2
Classpath	\${ORACLE_JDBC_DRIVER_PATH}/classes12.zip	\${SHADOW_JDBC_DRIVER_PATH}/scjd12.jar
Native Library	\${ORACLE_HOME}/lib	\${SHADOW_HOME}/lib



Path		
Implementation Classname	oracle.jdbc.pool.OracleConnectionPoolDataSource	com.neon.jdbc.DataSource

3.5.2 Oracle JDBC

For Oracle JDBC Thin Driver, create the following data sources (V5):

Name	JNDI Name	Custom Properties	Description	Connect Info
FOTW34TempAppDS	jdbc/FOTW34TempAppDS	<ul style="list-style-type: none">URL=jdbc:oracle:thin:@4.20.14.15:1666:FAFSAD34dataSourceName=FOTW34TempAppDSdatabaseName=FAFSAD34enableMultithreadedAccessDetection=falseportNumber=1666	FOTW 0304 Temp Save JDBC Datasource	Min=0 Max=10 Conn Timeout=1800 Aged Timeout=1800 Reap time=180
FOTWPin	jdbc/FOTWPin	<ul style="list-style-type: none">URL=jdbc:oracle:thin:@4.20.14.15:1692:EACDEVdataSourceName= FOTWPindatabaseName= EACDEVenableMultithreadedAccessDetection=falseportNumber=1692	PIN site Oracle JDBC Datasource	Min=0 Max=10 Conn Timeout=1800 Aged Timeout=1800 Reap time=180
FOTWTempAppDS	jdbc/FOTWTempAppDS	<ul style="list-style-type: none">URL=jdbc:oracle:thin:@4.20.14.15:1666:FAFSADEVdataSourceName=FOTW34TempAppDSdatabaseName=FAFSADEVenableMultithreadedAccessDetection=falseportNumber=1666	FOTW 0203 Temp Save JDBC Datasource	Min=0 Max=10 Conn Timeout=1800 Aged Timeout=1800 Reap time=180
FOTWSessions	Jdbc/FOTWSessions	<ul style="list-style-type: none">URL=jdbc:oracle:thin:@4.20.14.15:1666:FAFSADEVdataSourceName=FOTW34TempAppDSdatabaseName=FAFSADEVenableMultithreadedAccessDetection=falseportNumber=1666	FOTW Sessions DataBase	Min=0 Max=10 Conn Timeout=1800 Aged Timeout=1800 Reap time=180

The data source “Connect Info” should be used in production.

3.5.3 Shadow Direct JDBC

For ShadowDirect JDBC Driver, create the following data sources (V5):

Name	JNDI Name	Custom Properties	Description
FOTW12DataSource	jdbc/FOTW12DataSource	<ul style="list-style-type: none">dataSourceName=FOTWMF12Denable2Phase=false	0102 FOTW DB2 datasource (used by 0203 servlets)

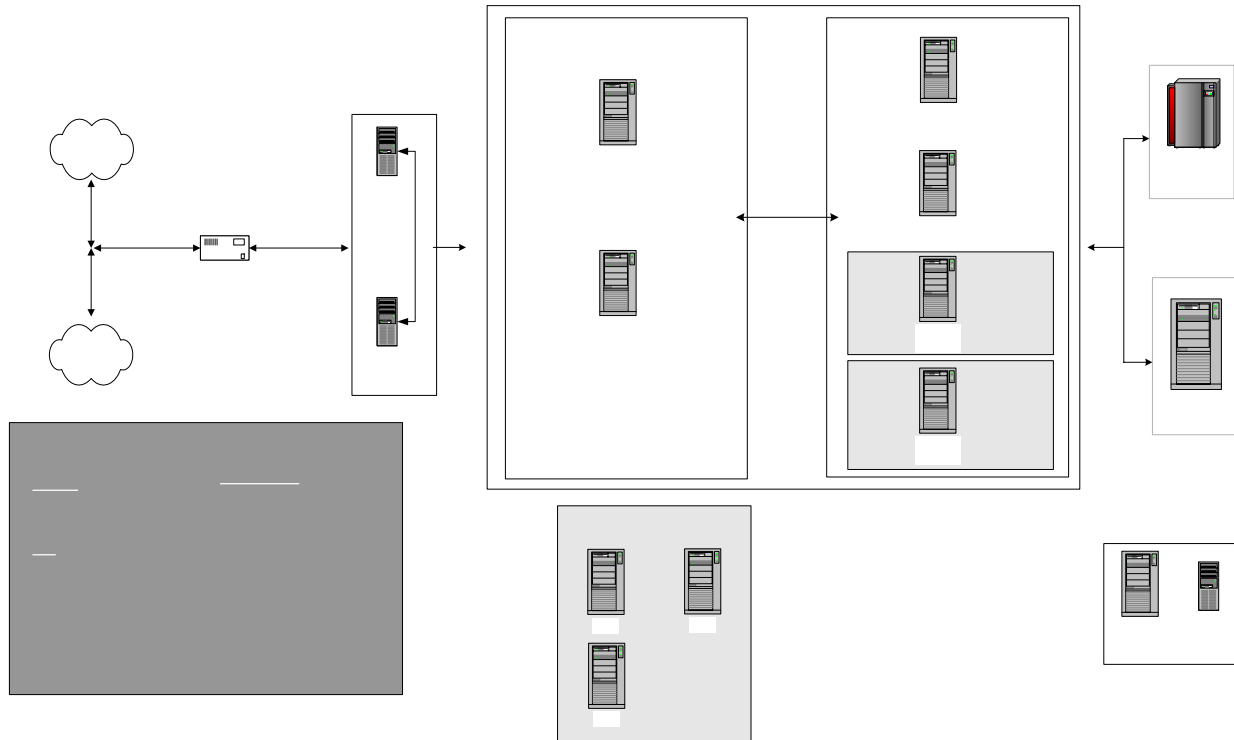


FAFSA Configuration Document

FOTW34DataSource	jdbc/FOTW34DataSource	<ul style="list-style-type: none">• dataSourceName=FOTWMF34D• enable2Phase=false	FOTW 0304 DB2 DataSource
FOTWDataSource	jdbc/FOTWDataSource	<ul style="list-style-type: none">• dataSourceName=FOTWMF23D• enable2Phase=false	0203 FOTW DB2 datasource
T4WANDDataSource	jdbc/T4WANDDataSource	<ul style="list-style-type: none">• dataSourceName= T4WANT• enable2Phase=false	T4Wan DB2 DataSource for FAA Access



4 Appendix – FAFSA Production Architecture



FSA ITA Release 4 FAFSA Production Architecture Physical Diagram As of 07/01/2003

4.1 Load Balancing

- eNetwork Dispatcher - SU22e1
- eNetwork Dispatcher - SU22e2

4.2 Web Servers

- IBM HTTP Server (IHS) 1.3.26.1 – HPN13
- IBM HTTP Server 1.3.26.1 – HPN14

Internet

4.3 Application Servers

- WebSphere Application Server (WAS) – HPN15
 - 2 clones – FAFSA
 - 1 clone – PIN
- WebSphere Application Server – HPN16
 - 2 clones – FAFSA
 - 1 clone – PIN
- WebSphere Application Server – HPN2
 - 2 clones – PIN

FW



- WebSphere Application Server – HPN4
 - 1 clone – FAFSA

4.4 Mainframe

CPS Production Region

4.5 Database

Oracle 8i DB server HPV2

NOTE: This document compliments a document that was recently distributed to the FAFSA team. The document is entitled “FOTW and PIN Configuration for WAS 5.0 Production” and it was supplied to CSC so that they were able to use that document to build the FAFSA 7 on WAS 5 production environment.